

REMARKS

Summary of Office Action

Claims 1 - 13, 15-27 and 29-41 are pending in the case. Claims 1, 16, and 30 are independent claims.

The Examiner rejected Claims 1-6, 8-10, 13, 15-21, 23-25, 29-35, and 37-39 under 35 U.S.C. § 102(a) as being anticipated by W3C, "Web Services Description Language (WSDL) 1.1," 03/15/01, pp. 1-51, <http://www.w3.org/TR/wsdl> (Hereafter "**W3C 1.1**").

The Examiner rejected Claims 7, 12, 22, 27, 36, and 40 under 35 U.S.C. § 103(a) as being unpatentable over W3C 1.1.

Claims 1 - 13, 15-27 and 29-41 remain in the application. Reconsideration of the rejection on the basis of the following remarks and analysis is requested.

Summary of Disclosure

The present disclosure is directed to a Type Description Language (TDL). The TDL is an extensible markup language (XML) based language that provides an interface description for mapping an interface specification to its wire format in a deterministic manner. The methodology enabled by the TDL represents the behavioral and data aspects of a type by creating a one to one mapping (injective function) from an abstract type to a schema. The TDL utilizes XML Schemas (XSD) enhanced to represent type references and arrays and numerous syntactic restrictions such as usage of element representation for fields as the canonical syntax to represent the data aspect of a type.

The TDL leverages the duality between the type-based (objects) and XML-based views and may be used for exchanging metadata between various kinds of type (object) systems, such as Component Object Model (COM), Common Object Request Broker Architecture (CORBA), Common Language Runtime (CLR), etc.

BASIC COMPUTER CONCEPTS

A. Application Interface

An Application Programming Interface (API) is an interface that a computer system, library or application (Application) provides to allow requests for services from other computer programs to be processed by the Application. The objective of an API is to enable software developers to access the functions or libraries of an Application without requiring access to the source code of the Application.

B. Services

A service is a software system designed to support inter operable machine to machine interaction over a network. The service will typically have an interface described in a machine processable format. Systems interact with a service in the mode specified by its interface using messages. Typically the message would be conveyed in XML and may be enclosed in a SOAP envelope. Applications written in different languages and operating on different platforms may use Web services to exchange data over computer networks.

C. XML Schema

An XML schema is a description of an XML document expressed in terms of constraints on the structure and content of the document. Schemas provide the sets of rules that define structure, content and semantic of XML documents. An XML Schema is a replacement, more complete of doctype. An XML schema provides a view of the document type at a relatively high level of abstraction. The mechanism for associating an XML document with a schema varies according to the schema language. The association may be achieved via markup within the XML document itself, or via some external means. The process of checking to see if an XML document conforms to a schema is called validation. Documents are only considered valid if they satisfy the requirements of the schema with which they have been associated.

D. One to One Mapping

A One to One Mapping (injective function) refers to a mapping that connects the members of a set A with members of another set B in a way that a single element of B is associated with each element of A, and no two elements of A map onto the same element of B.

The Cited Reference

The W3C 1.1 reference cited and applied by the Examiner deals with an XML grammar for describing network services as collections of communication endpoints capable of exchanging messages. The reference provides:

“A WSDL document defines services as collections of network endpoints, or ports. In WSDL the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service.” (W3C 1.1, page 3, Introduction).

The Claims

CLAIM 1

A. Substance of the Examiner's rejection

The Examiner rejected claim 1, under 35 USC § 102 (a) as being anticipated by W3C 1.1. The Examiner stated in the rejection:

"In regard to substantially similar independent claims 1, 16, 30 and dependent claims 13, 15, and 29, W3C 1.1 teaches a method, computer readable medium, and device for providing interface description for a service of a device in a computing system, comprising:

creating a one to one mapping of each type in the device or object to an XML schema (Page 4: "Types- a container for data type definitions using some type of system (such as XSD)"& "WSDL recognizes the need for rich type systems for describing message formats, and supports the XML schema specification"; Page 5: e.g. Example 1); and

describing the one to one mapping with an extensible markup language (XML)-based Interface Description Language (IDL) (Page 1 : Abstract; Pages 3 - 4: Introduction: "A WSDL document")."

B. W3C 1.1 does not disclose the element of creating a one to one mapping of each type in the device or object to an XML schema.

There is no disclosure in the cited reference of the step of creating a one to one mapping of each type of the device to an XML schema. Indeed the cited reference describes the concept of "Type" as used in the WSDL as follows:

"The **types** element encloses data type definitions that are relevant for the exchanged messages. For maximum interoperability and platform neutrality, WSDL prefers the use of XSD as the canonical type system, and treats it as the intrinsic type system.

```
<definitions .... >
  <types>
    <xsd:schema .... />*
  </types>
</definitions>
```

The XSD type system can be used to define the types in a message regardless of whether or not the resulting wire format is actually XML, *or whether the resulting XSD schema validates the particular wire format.*" (Emphasis added, W3C 1.1, p. 13)

As the Applicant explains in the Background of the invention in the Specification:

"In the ideal distributed computing environment described above, a computing device is not concerned with wire format. Indeed, ideally, the interface definition language used to drive the standardization of interface definitions for services also serves to drive the wire format utilized. Thus, there is also a need in the art for an interface description language that also serves as a wire format for standard exchange of interface definitions among computing devices."

As described in the specification, Applicant's Type Description Language ("TDL") operates in a different manner, namely:

"Fig. 2 captures the essence of the duality achieved by TDL between Object based and XML based views. Fig. 2 illustrates that there is a one to one mapping from an abstract type 200 to a Schema type 210 and vice-versa along pathway 205 in accordance with the present invention. There is also a one to one mapping from an instance 220 to an XML document 230 and vice-versa via a SOAP serializer 235 along pathway 235. The Is Instance operator along pathway 215 between an abstract type 200 and an instance 220 returns TRUE if and only if the Is Valid operator along pathway 225 returns TRUE between the corresponding XML Schema Type and XML Document. TDL is the first interface description language that ensures that both the Is Instance operator and Is Valid operator will return TRUE."

Figure 2 is reproduced below for the Examiner's convenience.

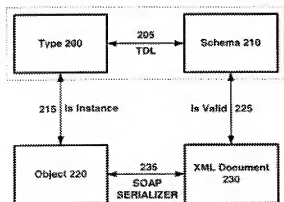


FIG. 2

Thus, unlike WSDL described in W3C 1.1, Applicant's one-to-one mapping results in an XML Schema that validates the wire format. (see reference 225 in Fig. 2).

Indeed, one of the main drawbacks of the Web Services Description Language (WSDL) 1.1 is that it is deficient in its interoperability because it does not provide one to one mapping between each type in the device or object to an XML schema. WSDL does not support semantic description of services. WSDL does not support the definition of logical constraints between its input and output parameters. In a detailed critique of the deficiencies of WSDL entitled "A Framework for Next Generation Enterprise Application Integration," Andrew Roszko, 2004, <http://ctd.uwaterloo.ca/ctd/aprroszko2004.pdf>. (See Appendix A), the author pointed out:

"Before this methodology can come to fruition, however, it is evident that the current interoperability issues of the web services model must be rectified. To this end, we feel that the concepts employed in the WSDL specification are sufficiently flawed to merit the proposal of several key amendments. The excessively complex, verbose WSDL spec mandates the application of an extensive, convoluted, contradictory set of rules for SOAP message creation. As a result, messages generated by one toolkit are often not recognized or even worse, are processed incorrectly by another. In order to achieve interoperability, XML messages must, quite simply, mean the same thing to every endpoint; servers must essentially be afforded the ability to validate incoming requests at the message level. *To this end, WSDL documents must therefore provide a schema definition for precisely what must appear on the wire; this simplistic approach removes any ambiguity and introduces instant interoperability into the platform.* This paradigm shift can be achieved with several essential changes to the WSDL spec, including mandating the use of XML Schema, the removal of the <wsdl:message> construct, as well as the elimination of both the 'encoded' and 'rpc' binding options." (Emphasis added).

Indeed, the only reference to mapping in W3C 1.1 is in pages 16-17 wherein it states:

“2.3.2 Abstract vs. Concrete Messages. Message definitions are always considered to be an abstract definition of the message content. A message binding describes how the abstract content is mapped into a concrete format. However, in some cases, the abstract definition Web Service Definition Language (WSDL) Page 16 of 51 <http://www.w3.org/TR/wsd15/8/2007> may match the concrete representation very closely or exactly for one or more bindings, so those binding(s) will supply little or no mapping information. However, another binding of the same message definition may require extensive mapping information. For this reason, it is not until the binding is inspected that one can determine "how abstract" the message really is.”

C. W3C 1.1 does not disclose the element of describing the one to one mapping with an extensible markup language (XML)-based Interface Description Language

An interface description language is a computer language used to describe a software component's interface. For example, objects in the CORBA distributed object environment are defined by an IDL, which describes the services performed by the object and how the data are to be passed to it. An IDL describes programming interfaces in a language neutral way and is used by tools to statically generate or dynamically configure interfaces, proxies, and ties in a specific environment.

The Web Services Description Language is an XML-based language that provides a model for describing Web services. WSDL does not describe everything about a service.

WSDL 1.1 only describes three things about a service:

- what it does: (this includes the interfaces (portTypes), supported operations, input and output messages for each, and the schema of those messages;
- how to communicate with it: (the bindings of each operation to specific protocols and encoding formats); and
- where to find it: (i.e. the endpoints for each binding).

WSDL does not specify the constraints and capabilities of the service nor the interaction semantics. WSDL does not define message semantics. Thus WSDL does not describe the one to one mapping with an extensible markup language (XML)-based Interface Description Language. The fact that WSDL is not an interface description language is made buttressed by the fact that an IDL (WSCl) for use with WSDL was described in Web Service Choreography Interface (WSCl) 1.0 W3C 1.1 Note 8 August 2002. (See Appendix B.)

It is respectfully submitted that Claim 1 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 1 is requested.

CLAIM 2

The Examiner rejected Claim 2 under 35 USC § 102(a) as anticipated by W3C 1.1. The examiner stated:

In regard to dependent claims 2, 17, and 31, W3C teaches wherein the XML based IDL is Type Description Language (TDL)(Page 4: "Types"; Pages 13-14: "2.2 Types").

However, WSDL is not a Type Description Language. WSDL is a description language for web service interfaces and a service implementation definition language. Applicant's TDL has the following characteristics (see Specification, p. 20, line 15 – p. 22, line 8):

- TDL includes sufficient information on all the parts of the action signature and supports subtyping.
- TDL abstracts the first-class concepts of certain distributed environments as first-class primitives.
- TDL enables the ability to state the intention of the action and also distinguish between various actions because the rules for ambiguity occurrence and resolution in TDL are clearly stated as part of the language definition. TDL also allows a single syntactic form for any semantic element.
- TDL may be extended to express the semantics of any specific type system.
- TDL enables the specification of parts of the service description independently and includes them, as needed.

WSDL is not a TDL as described in applicant's specification. Below is a partial comparison of different concepts in Applicant's TDL and WSDL as described in W3C 1.1.

Concept	Applicant' TDL	W3C 1.1
Service	A service is a set of interfaces where each interface can itself be a collection of actions, properties and event sources. (spec., p. 21, line 26)	A WSDL document defines services as collections of network endpoints, or ports. (W3C 1.1., p. 3)
Elements	actions, services, interfaces, methods, properties and event sources (Spec., p. 30 line 42 – p. 31, line 1)	Types Message Operation Port Type Binding Port Service (W3C 1.1, p 4)
Service Element	A service element represents the actions of a concrete entity, which could be a software application or a device. The service element is a named collection of interfaces, methods, properties and event sources that the clients of the service can use. (Spec. p. 31, line 4)	a collection of related endpoints. (W3C 1.1, p. 4)

Binding	For example, while deciding the constructs for protocol binding, TDL may mandate generic information, while leaving the protocol details to other specifications. (Spec, p. 25, line 5)	A binding defines message format and protocol details for operations and messages defined by a particular portType. There may be any number of bindings for a given portType. (W3C 1.1, p. 4)
---------	---	---

The W3C 1.1 reference does not describe a TDL as defined in applicant's claims. It is respectfully submitted that Claim 1 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 1 is requested.

CLAIM 3

The Examiner rejected Claim 3, under 35 USC § 102 (a) as anticipated by W3C 1.1. The examiner stated:

"... W3C teaches creating a one to one mapping from a programming construct (Page 5: Example 1 : "<types></types>") to an XML schema for describing the programming construct (Page 4: "WSDL recognizes the need for rich type systems for describing message formats, and supports the XML schema specification"; Page 9: "types, which provides data type definitions used to describe the messages exchanged").

Although the words "type" and "XML schema" appear on W3C 1.1, there is no indication or suggestion of creating a one to one mapping from a programming construct to an XML schema for describing the programming construct.

It is respectfully submitted that Claim 3 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 3 is requested.

CLAIM 4

The Examiner rejected claim 4 under 35 USC §102 (a), asserting that W3C 1.1 “teaches wherein the programming construct is one of pointer, class, array, subtype, enumeration, service reference, or bit field (Pages 13-14: “2.2 Types”).” However, there is no reference in the cited art to a method element of creating a one to one mapping of the programming construct as required by claim 3 on which claim 4 depends.

It is respectfully submitted that Claim 4 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 4 is requested.

CLAIM 5

The Examiner rejected claim 4 under 35 USC §102 (a), asserting that W3C 1.1 “teaches creating a one to one mapping from a constant value of complex type to an XML schema for describing the constant value of complex type (Page 11 : “<complexType> . . .</complexType>”).”

There is no disclosure in the reference of the concept of one to one mapping.

It is respectfully submitted that Claim 5 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 5 is requested.

CLAIM 6

The Examiner rejected claim 6 under 35 USC §102 (a), asserting that W3C 1.1 teaches “creating a one to one mapping from at least properties, methods, events of the type system to an XML schema for describing the at least one of properties, methods, events (Page 5: Example 1: “<element name= “tickerSymbol1” type= “stringw/>”

There is no disclosure in the reference of the concept of one to one mapping. See discussion regarding Claim 1.

It is respectfully submitted that Claim 6 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 6 is requested.

CLAIM 7

The Examiner rejected claim 6 under 35 USC §103 (a) asserting that “it would have been obvious to It would have been obvious to one of ordinary skill in the art at the time of the invention for the TDL of W3C to have supported inheritance of programming constructs, because W3C taught a TDL utilizing XML Schema, which was notoriously well known in the art at the time of the invention to provide inheritance to the typed programming constructs.” Applicant disagrees. Interestingly, the cited reference does not mention the concept of inheritance. However, in the new version of WSDL (WSDL 2.0, working draft published 26 march 2007, Appendix C), there is significant attention paid to the concept. Among the differences between WSDL 1.1 and WSDL 2.0 are Version 2.0 of WSDL (Web Services Description Language) a new component model, interface inheritance and other changes designed to reduce complexity. This raises the question that if it was obvious, why did it take 5 years to come up with the new version.

It is respectfully submitted that Claim 7 is not rendered obvious by W3C 1.1. Reconsideration and allowance of Claim 7 is requested.

CLAIM 8

The examiner rejected claim 8 under 35 USC §102 (a), asserting that W3C 1.1 teaches the XML-based IDL as a wire format for message communications relating to the service between devices of the computing system (Page 12: "wire format is actually XML). It is unclear from the cited page that the wire format is actually XML. Interestingly, page 13 of the reference includes the following:

“The XSD type system can be used to define the types in a message regardless of whether or not the resulting wire format is actually XML, or whether the resulting XSD schema validates the particular wire format.”

It therefore appears from the cited reference that there is a teaching away from the claimed subject matter.

It is respectfully submitted that Claim 8 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 8 is requested.

CLAIM 9

The examiner rejected claim 8 under 35 USC §102 (a), asserting that W3C 1.1 teaches “creating a one to one mapping from the wire format to the message communications (Page 12: “wire format is actually XML).” W3C 1.1 does not teach a one to one mapping from the wire format to the message communication as recited in claim 9. See discussion regarding Claim 8.

It is respectfully submitted that Claim 9 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 9 is requested.

CLAIM 10

The examiner rejected claim 10 under 35 USC §102 (a), asserting that W3C 1.1 teaches “TDL enables a transfer of a service reference across an application boundary (Page 1 : Abstract; Pages 3-4: Introduction).” There is no mention of a service reference in the W3C1.1 document. Nor is there any mention of a service reference that is transferred across an application boundary.

It is respectfully submitted that Claim 10 is not anticipated by W3C 1.1. Reconsideration and allowance of Claim 10 is requested.

CLAIM 11

The Examiner rejected Claim 11 under 35 USC § 103 over W3C 1.1, in view of Jeff Schneider, "Convergence of Peer and Web Services", 07/20/01, pp. 1-7, <http://www.openp2p.com/pub/a/p2p/2001/0720/convergence.html> (“Schneider”). The Examiner cited Schneider as teaching the eventual convergence of web services computing environment and a peer to peer environment. Respectfully, it is clear that Schneider does not teach but rather speculates about a possible convergence by means presumable to be invented in the future.

Neither Schneider, nor W3C 1.1 teach the element of creating a one to one mapping of each type in the device or object to an XML schema for use in a distributed computer environment.

It is respectfully submitted that Claim 11 is not rendered obvious by W3C 1.1 in combination with Schneider. Reconsideration and allowance of Claim 11 is requested.

CLAIM 12

The Examiner rejected Claim 12 under 35 USC § 103 (a) as being unpatentable over W3C 1.1. The Examiner stated:

“W3C does not specifically teach wherein the XML-based IDL was extendable to map additional constructs of a richer type system to an XML schema. It would have been obvious to one of ordinary skill in the art at the time of the invention for the XML based TDL of W3C to be extendable to map additional constructs or a richer type, because W3C taught a TDL utilizing XML Schema, which was notoriously well known in the art at the time of the invention to provide the extension element which allowed the appending of additional elements to an existing simpleType or complexType element construct.”

However, as laid out with regard to the rejection of Claim 1, W3C 1.1 fails to disclose the element of creating a one to one mapping of each type in the device or object to an XML schema. Furthermore, although the concept of extensibility was a known concept, achieving the extensibility by extending the interface description language to express the semantics of any specific type system as set out in applicant's specification is not disclosed by W3C 1.1. Consequently, the reference and the knowledge imputed by the Examiner does not render the claim obvious.

It is respectfully submitted that Claim 11 is not rendered obvious by W3C 1.1 in combination with Schneider. Reconsideration and allowance of Claim 11 is requested.

CLAIM 13

The Examiner rejected Claim 13 on the same basis as Claim 1. It is respectfully submitted that Claim 13 is allowable for the same reasons as asserted for Claim 1. Reconsideration and allowance of Claim 13 is requested.

CLAIM 15

The Examiner rejected Claim 15 on the same basis as Claim 1. It is respectfully submitted that Claim 15 is allowable for the same reasons as asserted for Claim 1. Reconsideration and allowance of Claim 15 is requested.

CLAIM 16

The Examiner rejected Claim 16 on the same basis as Claim 1. It is respectfully submitted that Claim 16 is allowable for the same reasons as asserted for Claim 1. Reconsideration and allowance of Claim 16 is requested.

CLAIM 17

The Examiner rejected Claim 17 on the same basis as Claim 2. It is respectfully submitted that Claim 17 is allowable for the same reasons as asserted for Claim 2. Reconsideration and allowance of Claim 17 is requested.

CLAIM 18

The Examiner rejected Claim 18 on the same basis as Claim 3. It is respectfully submitted that Claim 18 is allowable for the same reasons as asserted for Claim 3. Reconsideration and allowance of Claim 18 is requested.

CLAIM 19

The Examiner rejected Claim 19 on the same basis as Claim 4. It is respectfully submitted that Claim 19 is allowable for the same reasons as asserted for Claim 4. Reconsideration and allowance of Claim 19 is requested.

CLAIM 20

The Examiner rejected Claim 20 on the same basis as Claim 5. It is respectfully submitted that Claim 20 is allowable for the same reasons as asserted for Claim 5. Reconsideration and allowance of Claim 20 is requested.

CLAIM 21

The Examiner rejected Claim 21 on the same basis as Claim 6. It is respectfully submitted that Claim 21 is allowable for the same reasons as asserted for Claim 6. Reconsideration and allowance of Claim 21 is requested.

CLAIM 22

The Examiner rejected Claim 22 on the same basis as Claim 7. It is respectfully submitted that Claim 22 is allowable for the same reasons as asserted for Claim 7. Reconsideration and allowance of Claim 22 is requested.

CLAIM 23

The Examiner rejected Claim 23 on the same basis as Claim 8. It is respectfully submitted that Claim 23 is allowable for the same reasons as asserted for Claim 8. Reconsideration and allowance of Claim 23 is requested.

CLAIM 24

The Examiner rejected Claim 24 on the same basis as Claim 9. It is respectfully submitted that Claim 24 is allowable for the same reasons as asserted for Claim 9. Reconsideration and allowance of Claim 24 is requested.

CLAIM 25

The Examiner rejected Claim 25 on the same basis as Claim 10. It is respectfully submitted that Claim 25 is allowable for the same reasons as asserted for Claim 10. Reconsideration and allowance of Claim 25 is requested.

CLAIM 26

The Examiner rejected Claim 26 on the same basis as Claim 11. It is respectfully submitted that Claim 26 is allowable for the same reasons as asserted for Claim 11. Reconsideration and allowance of Claim 26 is requested.

CLAIM 27

The Examiner rejected Claim 27 on the same basis as Claim 12. It is respectfully submitted that Claim 27 is allowable for the same reasons as asserted for Claim 12. Reconsideration and allowance of Claim 27 is requested.

CLAIM 29

The Examiner rejected Claim 29 on the same basis as Claim 1. It is respectfully submitted that Claim 29 is allowable for the same reasons as asserted for Claim 1. Reconsideration and allowance of Claim 29 is requested.

CLAIM 30

The Examiner rejected Claim 30 on the same basis as Claim 1. It is respectfully submitted that Claim 30 is allowable for the same reasons as asserted for Claim 1. Reconsideration and allowance of Claim 30 is requested.

CLAIM 31

The Examiner rejected Claim 31 on the same basis as Claim 2. It is respectfully submitted that Claim 31 is allowable for the same reasons as asserted for Claim 2. Reconsideration and allowance of Claim 31 is requested.

CLAIM 32

The Examiner rejected Claim 32 on the same basis as Claim 3. It is respectfully submitted that Claim 32 is allowable for the same reasons as asserted for Claim 3. Reconsideration and allowance of Claim 32 is requested.

CLAIM 33

The Examiner rejected Claim 33 on the same basis as Claim 4. It is respectfully submitted that Claim 33 is allowable for the same reasons as asserted for Claim 4. Reconsideration and allowance of Claim 33 is requested.

CLAIM 34

The Examiner rejected Claim 34 on the same basis as Claim 5. It is respectfully submitted that Claim 34 is allowable for the same reasons as asserted for Claim 5. Reconsideration and allowance of Claim 34 is requested.

CLAIM 35

The Examiner rejected Claim 35 on the same basis as Claim 6. It is respectfully submitted that Claim 35 is allowable for the same reasons as asserted for Claim 6. Reconsideration and allowance of Claim 35 is requested.

CLAIM 36

The Examiner rejected Claim 36 on the same basis as Claim 7. It is respectfully submitted that Claim 36 is allowable for the same reasons as asserted for Claim 7. Reconsideration and allowance of Claim 36 is requested.

CLAIM 37

The Examiner rejected Claim 37 on the same basis as Claim 8. It is respectfully submitted that Claim 37 is allowable for the same reasons as asserted for Claim 8. Reconsideration and allowance of Claim 37 is requested.

CLAIM 38

The Examiner rejected Claim 38 on the same basis as Claim 9. It is respectfully submitted that Claim 38 is allowable for the same reasons as asserted for Claim 9. Reconsideration and allowance of Claim 38 is requested.

CLAIM 39

The Examiner rejected Claim 39 on the same basis as Claim 10. It is respectfully submitted that Claim 39 is allowable for the same reasons as asserted for Claim 10. Reconsideration and allowance of Claim 39 is requested.

DOCKET NO.: MSFT-0736/183220.01

Application No.: 10/017,265

Office Action Dated: March 8, 2007

PATENT

CLAIM 40

The Examiner rejected Claim 40 on the same basis as Claim 12. It is respectfully submitted that Claim 40 is allowable for the same reasons as asserted for Claim 12. Reconsideration and allowance of Claim 40 is requested.

CLAIM 41

The Examiner rejected Claim 41 on the same basis as Claim 11. It is respectfully submitted that Claim 41 is allowable for the same reasons as asserted for Claim 11. Reconsideration and allowance of Claim 41 is requested.

DOCKET NO.: MSFT-0736/183220.01

Application No.: 10/017,265

Office Action Dated: March 8, 2007

PATENT

CONCLUSION

Applicants believe that the present Amendment is responsive to each of the points raised by the Examiner in the Office Action, and submit that Claims 1-13, 15-27 and 29-41 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Respectfully submitted,

Date: June 7, 2007

/Eduardo M. Carreras/
Eduardo M. Carreras
Registration No. 28,197

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439